

## SERIAL BUS INTERFACE DEVICE

BACKGROUND OF THE INVENTION

## 1. Field of the Invention

The present invention relates to a serial bus interface device to be connected to a next-generation digital interface typified by the IEEE 1394 serial bus. More particularly, the invention relates to a serial bus interface device suitable as a bus analyzer for checking a communication state on a bus such as an IEEE 1394 serial bus, operations of a device connected to the bus, and the like.

## 2. Description of Related Art

A next-generation digital interface typified by the IEEE 1394 serial bus is used not only for data transfer between a personal computer and a peripheral device of the personal computer but also for transfer of multimedia data such as moving picture data between a digital AV device such as a digital camera or a DVD player and a personal computer and between digital AV devices. The digital interface is therefore requested to have a high data transfer speed. The application range of the digital interface includes ordinary households. Consequently, what is called a plug-and-play function which can deal with the insertion or withdrawal of a device in a power-on state is necessary.

The IEEE 1394 serial bus has a high data transfer speed such that a packet transfer speed is 100 to 400 megabits per second and supports two kinds of transfer modes: a synchronous transfer mode for stable transfer of multimedia data; and an asynchronous transfer mode for transferring conventional-type data and the like. In order to realize the plug-and-play function, the IEEE 1394 serial bus has a function of reconstructing the topology by executing an initial sequence when a device is inserted or withdrawn, and automatically assigning an identification number (ID) to each device.

For example, as shown in Fig. 1, in order to analyze the communication state between a personal computer 101 (ID = 1) and a digital camera 102 (ID = 0) which are connected via an IEEE 1394 serial bus B, it is necessary to allow a bus analyzer 100 to receive packets on the bus B. For this purpose, a construction that the bus analyzer 100 has a physical layer circuit compliant with the communication protocol of the IEEE 1394 standard is achieved by inserting the bus analyzer 100 on the bus B connecting the personal computer 101 and the digital camera 102. Specifically, the personal computer 101 and the bus analyzer 100 are connected via a bus B1 and the bus analyzer 100 and the digital camera 102 are connected via a bus B2. The IEEE 1394 serial bus reconstructs a bus topology, assigns a new ID (= 1) to the bus analyzer 100 via a physical layer circuit, and recognizes the bus analyzer 100 as a device connected to the buses B1 and B2. After the bus analyzer 100 is incorporated into the IEEE 1394 serial bus and the communication on the buses B1 and B2 becomes possible, the bus analyzer 100 analyzes the communication state between the personal computer 101 and the digital camera 102. In this case, since the ID is simultaneously reassigned to an existing device, the personal computer 101 is recognized as a device having the ID of 2 in place of the ID of 1.

Fig. 2 shows a case where  $(n + 1)$  devices are connected on the serial bus and a method of analyzing the communicating operation, for example, on a bus BB of a device A (ID = 0) in an IEEE 1394 serial bus system comprising devices A to H (IDs = 0 to 7) and other devices (IDs = 8 to n). A case of, for example, analyzing the communicating operation of the device A such as a newly-developed hard disk drive which is connected to the bus is assumed. The bus BB connecting the device A (ID = 0) and the device B (ID = 4) is divided into a bus BB1 and a bus BB2, and the bus analyzer 100 having a physical layer circuit

is inserted between the buses BB1 and BB2. In this case as well, in a manner similar to the case of Fig. 1, the IEEE 1394 serial bus reconstructs the bus topology, recognizes the bus analyzer 100 as a device having the ID of 1, resets the IDs of the other devices, and constructs an IEEE 1394 serial bus in a state where the bus analyzer 100 is incorporated. The communications on the buses BB1 and BB2 become possible and the bus analyzer 100 analyzes the communicating operation of the device A.

In order to analyze the communicating state between the personal computer 101 and the digital camera 102, however, the serial bus interface device used as the bus analyzer 100 shown in Fig. 1 has to be inserted in the bus B connecting the personal computer 101 and the digital camera 102 via the physical layer circuit. By inserting the bus analyzer 100, the bus analyzer 100 itself becomes one of the devices on the bus. Consequently, the bus configuration has a topology including the bus analyzer 100 which does not exist in an inherent connection environment. It causes a problem such that analysis in a communication environment in the inherent topology cannot be performed.

Further, in the serial bus interface device used as the bus analyzer 100 shown in Fig. 2, in a manner similar to the case of Fig. 1, by inserting the bus analyzer 100, the configuration of buses changes and it causes a problem such that analysis in a communication environment in the inherent topology cannot be performed.

When the communicating state of a bus has to be analyzed in a state where a number of devices are connected to the bus, it is necessary to connect all the devices so as to be adapted to a regular use environment and then analyze the communicating state. When the analysis has to be performed in a state where there are a number of devices to be connected or a number of kinds of devices, there is a

problem such that enormous efforts are required to set an analysis environment.

#### SUMMARY OF THE INVENTION

The invention has been achieved to solve the problems of the conventional arts and an object of the invention is to provide a serial bus interface device having a physical layer circuit, which is to be connected to a serial bus such as an IEEE 1394 serial bus, can be inserted or withdrawn without exerting an influence on the bus topology, can simulate a plurality of devices, and is suitably used as a bus analyzer for analyzing a communicating state of a bus.

In order to achieve the object, according to one aspect of the invention, a serial bus interface device having a function of automatically reconstructing a topology when the device is inserted or withdrawn during operation of a serial bus comprises a physical layer circuit serving as a physical interface without being given an identification number when the serial bus interface device is connected to the serial bus.

The serial bus interface device can be connected without changing the topology environment of existing devices connected to the serial bus and without exerting an influence on responses on the serial bus. The physical layer circuit of the serial bus interface device connected can serve as a physical interface with the serial bus, so that data on the serial bus can be detected.

According to another aspect of the invention, a serial bus interface device having a function of automatically reconstructing a topology when the device is inserted or withdrawn during operation of a serial bus comprises a physical layer circuit serving as a physical interface to which one or more identification numbers are assigned when the serial bus interface device is connected to the serial bus.

When the serial bus interface device is connected to the serial bus, the physical layer circuit can send responses in a manner similar to a case where all of one or more devices each having an identification number are connected. By the single serial bus interface device, the environment which is the same as that in the case where a number of devices are connected can be realized.

The above and further objects and novel features of the invention will more fully appear from the following detailed description when the same is read in connection with the accompanying drawings. It is to be expressly understood, however, that the drawings are purpose of illustration only and not intended as a definition of the limits of the invention.

#### BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 shows an IEEE 1394 bus configuration in which a conventional bus analyzer is connected.

Fig. 2 shows a configuration in which a conventional bus analyzer is connected to an IEEE 1394 bus to which a number of devices are connected.

Fig. 3 shows a configuration of an IEEE 1394 bus to which a bus analyzer of a first embodiment is connected.

Fig. 4 is a state transition diagram showing a tree-identifying operation in the first embodiment.

Fig. 5 is a state transition diagram showing a self-identifying operation in the first embodiment.

Fig. 6 shows an operation sequence in the case of receiving a parent\_notifying signal by one of ports in the tree-identifying operation in the first embodiment.

Fig. 7 shows an operation sequence in the case of receiving a parent\_notifying signal by both ports in the tree-identifying

operation in the first embodiment.

Fig. 8 shows an operation sequence of the self-identifying operation in the first embodiment.

Fig. 9 is a circuit block diagram showing a first modification of the bus analyzer in the first embodiment.

Fig. 10 is a circuit block diagram showing a second modification of the bus analyzer in the first embodiment.

Fig. 11 is a circuit block diagram showing a third modification of the bus analyzer in the first embodiment.

Fig. 12 is a circuit block diagram showing a fourth modification of the bus analyzer in the first embodiment.

Fig. 13 is a circuit block diagram showing a fifth modification of the bus analyzer in the first embodiment.

Fig. 14 is a circuit block diagram showing a sixth modification of the bus analyzer in the first embodiment.

Fig. 15 shows a first configuration example of an IEEE 1394 bus to which a bus analyzer of a second embodiment is connected.

Fig. 16 is a state transition diagram showing a self-identifying operation in the first configuration example of the second embodiment.

Fig. 17 shows an operation sequence of the case where arbitration of a device connected on the other side is higher in the self-identifying operation in the first configuration example of the second embodiment.

Fig. 18 shows an operation sequence of the case where arbitration of a device connected on the other side is lower in the self-identifying operation in the first configuration example of the second embodiment.

Fig. 19 shows a second configuration example of the IEEE 1394 bus to which the bus analyzer of the second embodiment is connected.

Fig. 20 is a state transition diagram showing a self-identifying operation in a second configuration example of the second embodiment.

Fig. 21 shows a front stage portion of an operation sequence in

the case where there is a device having high arbitration to be connected on the other side in the self-identifying operation in the second configuration example of the second embodiment.

Fig. 22 shows an intermediate stage portion of the operation sequence in the case where there is a device having high arbitration to be connected on the other side in the self-identifying operation in the second configuration example of the second embodiment.

Fig. 23 shows a post stage portion of the operation sequence in the case where there is a device having high arbitration to be connected on the other side in the self-identifying operation in the second configuration example of the second embodiment.

Fig. 24 shows a front stage portion of an operation sequence in the case where there is no device having high arbitration to be connected on the other side in the self-identifying operation in the second configuration example of the second embodiment.

Fig. 25 shows a post stage portion of the operation sequence in the case where there is no device having high arbitration to be connected on the other side in the self-identifying operation in the second configuration example of the second embodiment.

#### DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

A serial bus interface device according to an embodiment of the invention will be described in detail hereinbelow with reference to Figs. 3 to 25.

Fig. 3 shows the configuration of an IEEE 1394 bus to which a bus analyzer according to a first embodiment of the invention is connected. In order to analyze a communicating state on a bus in a system (not shown) in which the personal computer 101 having an identification number (hereinbelow, referred to as an ID) of 1 and the digital camera 102 having an ID of 0 are connected via an IEEE 1394

serial bus, a bus analyzer 1A of the invention is inserted between the personal computer 101 and the digital camera 102, the personal computer 101 and the bus analyzer 1A are connected via a bus B1, and the digital camera 102 and the bus analyzer 1A are connected via a bus B2. The bus analyzer 1A interfaces with the IEEE 1394 buses B1 and B2 by a physical layer circuit 2A in the bus analyzer 1A, thereby making communications with the buses possible. Received data is analyzed by a data analyzing circuit 3 at the post stage. As will be described hereinlater, even when the bus analyzer 1A is inserted, an ID is not assigned to the physical layer circuit 2A (having no ID) and, in such a state, the topology is reconstructed. The IDs of the personal computer 101 (ID = 1) and the digital camera 102 (ID = 0) therefore do not change before and after the insertion of the bus analyzer 1A. The communicating state can be analyzed without changing the bus topology of the system to be analyzed (in this case, the system constructed by the personal computer 101 (ID = 1) and the digital camera 102 (ID = 0)).

Referring to Figs. 4 to 8, the procedure of reconstructing the topology without assigning an ID to the physical layer circuit 2A in a sequence of initializing the IEEE 1394 bus in the physical layer circuit 2A when the bus analyzer 1A is inserted to the system to be analyzed will be described. Fig. 4 is a state transition diagram showing a tree-identifying operation of determining a tree structure of a topology on insertion of the bus analyzer 1A and reset of the bus. Each of Figs. 6 and 7 shows an actual operating sequence in the case of inserting the bus analyzer 1A into the system to be analyzed which comprises the personal computer 101 and the digital camera 102 in accordance with the state transition of Fig. 4. Fig. 5 is a state transition diagram showing a self-identifying operation of assigning IDs to devices after determining the tree structure. Fig. 8 shows an

operating sequence of assigning an ID to the system to be analyzed which comprises the personal computer 101 and the digital camera 102, to which the bus analyzer 1A is inserted in accordance with the state transition.

First, a case of receiving a parent\_notification signal by one of the ports of the physical layer circuit 2A in the tree-identifying operation will be described with reference to Figs. 4 to 6. The parent\_notification signal is a signal declaring that a port has high arbitration of the bus among ports connected to one of branches constructing the tree structure. When the bus analyzer 1A is inserted and the bus reset is completed, in a state (hereinbelow, referred to as "S") 1 in Fig. 4, according to a time-out sequence, an idle signal is periodically transmitted to notify that the physical layer circuit 2A in the bus analyzer 1A is in a stand-by state in the tree-identifying operation.

On condition that the parent\_notification signal is received by one of the ports of the physical layer circuit 2A in such a state (S2 in Fig. 4), a child\_notification signal is outputted to the same port (S3 in Fig. 4). It corresponds to procedures (hereinbelow, referred to as "P") 1 and 2 in the operating sequence of Fig. 6, in which on condition that the physical layer circuit 2A receives the parent\_notification signal outputted from the digital camera 102 to the serial bus B2 (P1), the child\_notification signal is outputted from the port toward the digital camera 102 (P2 in Fig. 6).

When the digital camera 102 which has received the child\_notification signal stops outputting the parent\_notification signal, a child\_handshake state is received (S4 in Fig. 4 and P3 in Fig. 6), thereby determining the tree structure between the physical layer circuit 2A and the digital camera 102. Subsequently, the physical layer circuit 2A outputs the parent\_notification signal to the other port (S5 in Fig. 4 and P4 in Fig. 6).

The operation of outputting the signal is continued until the device on the other side returns the child\_notification signal, the parent\_handshake state is received, and the tree structure is determined. That is, in Fig. 6, in the case where both the physical layer circuit 2A and the personal computer 101 output the parent\_notification signal onto the bus B1, a root\_contention state occurs according to the standard of the IEEE 1394 bus, and the personal computer 101 stops outputting the parent\_notification signal (P5 in Fig. 6). On the other hand, the physical layer circuit 2A keeps on outputting the parent\_notification signal (S5 in Fig. 4 and P5 in Fig. 6). After the time out of the personal computer 101, the personal computer 101 receives the parent\_notification signal. In response to the signal, the personal computer 101 outputs the child\_notification signal to the bus B1 and receives a parent\_handshake state (P6 in Fig. 6), thereby determining the tree structure and finishing the tree\_identifying operation (S6 in Fig. 4 and P12 in Fig. 6).

Although the mode in which the physical layer circuit 2A of the bus analyzer 1A responds to the signals received by each of the ports has been described as an example in the foregoing embodiment, the invention is not limited to the mode. It is also possible to determine the tree structure by sending a signal received by one of the ports to the other port.

A case of receiving the parent\_notification signal by both ports of the physical layer circuit 2A in the tree-identifying operation will now be described with reference to Figs. 4 to 7. In the following description, parts similar to those described above will not be described.

In the idle state (S1 in Fig. 4), when the parent\_notification signal is received from each of the personal computer 101 and the digital camera 102 by both ports of the physical layer circuit 2A (S2 in Fig.

4 and P7 in Fig. 7), the physical layer circuit 2A outputs the child\_notification signal to an arbitrary port (S3 in Fig. 4). In Fig. 7, the child\_notification signal is outputted toward the digital camera 102 (P8 in Fig. 7).

When the digital camera 102 which has received the child\_notification signal stops outputting the parent\_notification signal and the child\_handshake state is received (S4 in Fig. 4 and P9 in Fig. 7), the structure of the tree is determined and the parent\_notification signal is outputted to the other port toward the personal computer 101 (S5 in Fig. 4 and P9 in Fig. 7).

The following sequence is similar to the above-described case. To be specific, in the period from the receipt of the parent\_handshake state until the determination of the tree structure, the physical layer circuit 2A keeps on outputting the parent\_notification signal (S5 in Fig. 4 and P9 and P10 in Fig. 7). During the period of time, in Fig. 7, the parent\_notification signal from the physical layer circuit 2A and that from the personal computer 101 collide with each other, so that a root\_contention state is derived. Although the personal computer 101 stops outputting the parent\_notification signal (P9 and P10 in Fig. 7), the physical layer circuit 2A continuously outputs the parent\_notification signal (P10 in Fig. 7). After the time out, the personal computer 101 receives the parent\_notification signal and returns the child\_notification signal, thereby receiving the parent\_handshake state, determining the tree structure (P11 in Fig. 7) and finishing the tree-identifying operation (S6 in Fig. 4 and P12 in Fig. 7).

When the tree structure is determined by the tree-identifying operation, the arbitration of each of the ports connected to the bus is decided. More specifically, with respect to the bus B1 in Fig. 6, arbitration lower than that of the personal computer 101 is set to the

port of the personal computer 101, and arbitration higher than that of the bus analyzer 1A is set to the port of the bus analyzer 1A. With respect to the bus B2, arbitration higher than that of the digital camera 102 is set to the port of the digital camera 102, and arbitration lower than that of the bus analyzer 1A is set to the port of the bus analyzer 1A. In the following description, the port having high arbitration is defined as a parent port (it is described as "high" in Fig. 6 and subsequent drawings), and the port having low arbitration is defined as a child port (it is described as "low" in Fig. 6 and subsequent drawings). A device whose ports connected to a bus are all child ports is defined as a "root". In Figs. 6 and 7, the personal computer 101 is a root.

The self-identifying operation will now be described with reference to Figs. 5 and 8. After the tree-identifying operation is finished, in an idle state in which no signal is outputted from each of the ports (S7 in Fig. 5), when a self\_ID\_grant signal generated from the personal computer 101 as a root in order to retrieve a device to which the least significant ID is to be given is received by the parent port of the bus analyzer 1A (S8 in Fig. 5), the physical layer circuit 2A outputs the self\_ID\_grant signal to the child port of the bus analyzer 1A (S9 in Fig. 5 and P13 in Fig. 8).

When the device which receives the self\_ID\_grant signal and to which an ID is assigned (the digital camera 102 in Fig. 8 to which the ID of 0 is assigned) returns a self\_ID packet (self\_ID packet 0) and the child port of the bus analyzer 1A receives the self\_ID packet (S10 in Fig. 5 and P14 in Fig. 8), the self\_ID packet is transferred to the parent port (S11 in Fig. 5 and P14 in Fig. 8).

Further, when the child port receives an ident\_done signal from the digital camera 102 to which the ID of 0 is assigned (S12 in Fig. 5 and P15 in Fig. 8), the ident\_done signal and a speed signal indicative

of a data transfer speed received by the child port are transferred to the parent port (S13 in Fig. 5 and P15 in Fig. 8). The speed signal returned from the personal computer 101 to the parent port is transferred to the child port (S13 in Fig. 5 and P16 in Fig. 8), thereby determining the data transfer speed between the personal computer 101 and the digital camera 102.

When a self\_ID packet (self\_ID packet 1) from the personal computer 101 is received by the parent port (S14 in Fig. 5 and P17 in Fig. 8), the ID of the personal computer 101 as a device connected to the parent port side is also determined as "1", and the self-identifying operation is finished. In P17 in Fig. 8, the operation of transferring the self\_ID packet (self\_ID packet 1) to the child port is an operation executed in regular arbitration.

Each of the state transition from S7 to S11 in association with the reception of the self\_ID packet by the parent port in Fig. 5 and the state transition from S11 to S7 in association with the end of the self\_ID packet represents the function of performing the self-identifying operation without outputting the self\_ID\_grant signal. In this case as well, an ID is not assigned to the bus analyzer 1A.

As described above, in the case where the bus analyzer 1A of the first embodiment is inserted between the serial buses B1 and B2 of the system to be analyzed, since the bus analyzer 1A has the physical layer circuit 2A, the bus analyzer 1A can construct a physical interface between the buses B1 and B2 without being assigned an ID in the bus initializing sequence executed after the insertion. Without changing the topology environment of the system to be analyzed in which the personal computer 101 and the digital camera 102 are directly connected to the serial buses, the bus analyzer 1A as a serial bus interface device can be connected. Further, the physical layer circuit 2A of the bus analyzer 1A inserted can construct a physical interface with the buses

B1 and B2, detect data on the buses B1 and B2, and analyze and investigate the communicating state on the buses B1 and B2.

Bus analyzers 1A1 to 1A6 as first to sixth modifications of the first embodiment will now be described. Fig. 9 shows the first modification. The bus analyzer 1A1 has a data storing circuit 4 for storing data on the serial buses B1 and B2 received by the physical layer circuit 2A1. The bus analyzer 1A1 can construct a physical interface with the buses B1 and B2 without being given an ID. Consequently, the data on the buses B1 and B2 can be acquired without changing the topology environment of the system to be analyzed and without exerting an influence on the communicating operation. The data can be directly transferred to a data analyzing circuit 31 and analyzed. In addition, data can be properly stored and read to/from the data storing circuit 4 in accordance with a control signal from the data analyzing circuit 31. A series of continuous data or data in a specific sequence can be therefore collected and analyzed. The data can be also analyzed by comparison with data directly transferred from the physical layer circuit 2A1. Thus, the analysis can be effectively performed.

Fig. 10 shows a second modification. The bus analyzer 1A2 has, in addition to the data storing circuit 4, a data condition detecting circuit 5. The data condition detecting circuit 5 monitors data received by the physical layer circuit 2A2 and, when data matching a predetermined condition is detected, outputs a trigger signal. The bus analyzer 1A2 can store data received in response to the output of the trigger signal into the data storing circuit 4 and has a function of performing an operation on the stored data in a manner similar to the first modification. By setting the data condition on which the trigger signal is outputted in a specific data sequence, data can be stored each time the same sequence is generated. Consequently, analysis suitable for the case where a malfunction occurs under a

specific condition can be performed.

Fig. 11 shows a third modification. A bus analyzer 1A3 has, in addition to a data analyzing circuit 33, a data transfer control circuit 6. The data transfer control circuit 6 performs a control of transferring data to be transmitted as it is to the physical layer circuit 2A3 or forming data into a packet compatible with the protocol on the serial bus and transferring the packet to the physical layer circuit 2A3. The modification is suited to the case of recognizing a response on the bus to a predetermined data packet.

Fig. 12 shows a fourth modification. A bus analyzer 1A4 has, in addition to a data analyzing circuit 34 and the data transfer control 6, the data storing circuit 4. The data storing circuit 4 can prestore data to be transmitted and performs a control of transferring the data as it is or forming data in a packet compatible with the protocol on the serial bus and then transferring the packet to the physical layer circuit 2A4 as necessary. By storing plural data, each of responses on the serial bus to each of data packets can be recognized. Further, by arbitrarily setting a transmission sequence of data packets, a response to a specific packet sequence or a response in a state where communications on the bus are congested can be also recognized.

Fig. 13 shows a fifth modification. A bus analyzer 1A5 has, in addition to a data analyzing circuit 35, the data storing circuit 4, and the data transfer control circuit 6, a data transmission condition detecting circuit 7. The data transmission condition detecting circuit 7 monitors data received by the physical layer circuit 2A5 and, when data matching a predetermined condition is detected, outputs a trigger signal. On output of the trigger signal, the bus analyzer 1A5 performs a control of transferring data to be transmitted as it is which is stored in the data storing circuit 4 to the physical layer circuit 2A5 or forming the data to be transmitted into a packet compatible with

the protocol on the serial bus and transferring the packet to the physical layer circuit 2A5. By setting a condition of data to be received upon which the trigger signal is outputted in a specific data sequence, each time the same sequence is generated, predetermined data in the data stored in the data storing circuit 4 can be transmitted. The same response can be therefore always sent to the specific data sequence. Thus, the modification is the optimum to recognize a response on the serial bus.

Fig. 14 shows a sixth modification. A physical layer circuit 2A6 has a pair of ports, a receiving circuit 11, a data converting circuit 10 for converting data included in a packet received by the receiving circuit 11, and a transmitting circuit 12 for forming a packet from the data converted by the data converting circuit 10. The ports of transmission and reception are switched by a selector 13. On the basis of either the analysis of reception data by a data analyzing circuit 36 or the data to be transmitted which is stored in the data storing circuit 4, data received by the receiving circuit 11 in the physical layer circuit 2A6 is converted by the data converting circuit 10. In the modification, a communication error such as a bit error or a burst error in the communication on the serial bus can be simulated, and a communication response of an error correcting function or the like can be recognized.

A second embodiment will now be described. Fig. 15 shows a first configuration example of an IEEE 1394 bus to which a bus analyzer of the second embodiment of the invention is connected. The first configuration example relates to a case where a group of devices having a plurality of IDs are simulated by a bus analyzer 1B. It is intended for analyzing or investigating a response on a serial bus B3 of a personal computer 103 (or digital camera 103) in the case where  $(n + 1)$  devices having IDs 0 to  $n$  (or  $n$  devices having IDs 1 to  $n$ ) are connected

to the personal computer 103 having an ID of  $(n + 1)$  (or the digital camera 103 having an ID of 0). For analysis and investigation, it is not necessary to actually construct a system to be analyzed by preparing  $(n + 1)$  devices having IDs 0 to  $n$  (or  $n$  devices having IDs 1 to  $n$ ), a connection environment which is the same as that of a system to be analyzed can be created by connecting the bus analyzer 1B and the personal computer 103 (or digital camera 103) via a serial bus B3. The bus analyzer 1B interfaces with an IEEE 1394 bus B3 in a physical layer circuit 2B1 in the bus analyzer 1B and realizes data transmission/reception to/from the bus B3. Received data is analyzed by the data analyzing circuit 3 at the post stage. In the interface, the topology is constructed by assigning a plurality of IDs to the physical layer circuit 2B1 as will be described hereinlater. Consequently, only by connecting the personal computer 103 (or digital camera 103) to the bus analyzer 1B, an environment in which a plurality of devices are connected can be created.

The initializing sequence of the IEEE 1394 bus in the first configuration example of the second embodiment in Fig. 15 will now be described hereinbelow. Since the tree structure is determined in conformity with the IEEE 1394 standard, description of the tree-identifying operation of determining the tree structure of the topology is omitted here. A self-identifying operation of assigning IDs to devices will be described. Fig. 16 is a state transition diagram showing the self-identifying operation in the first configuration example of the second embodiment in Fig. 15. Fig. 17 shows an operation sequence of the case where the device to which the bus analyzer 1B is connected is the personal computer 103 (ID =  $n + 1$ ) having high arbitration. Fig. 18 shows an operation sequence of the case where the device to which the bus analyzer 1B is connected is the digital camera 103 (ID = 0) having low arbitration.

The case where the device to which the bus analyzer is connected is the personal computer 103 will be described first with reference to Figs. 16 and 17. In Fig. 16, after completion of the tree-identifying operation, an idle state (S21 in Fig. 16) in which no signal is outputted from each of the ports is derived. After that, the personal computer 103 as a root in the topology outputs the self\_ID\_grant signal to its child port. On condition that the self\_ID\_grant signal is received by the parent port (S22 in Fig. 16 and P21 in Fig. 17), the physical layer circuit 2B1 in the bus analyzer 1B which is not a root outputs a self\_ID packet (self\_ID packet 0) to the parent port to set the ID of 0 and increments an ID counter by one (S23 in Fig. 16 and P22 in Fig. 17).

The sequence is repeated until the ID counter becomes n (S24 in Fig. 16) and, subsequent to the ID of 0, IDs 1 to n are sequentially set in the physical layer circuit 2B1 (P23, P24 and P25 in Fig. 17).

When the ID counter becomes n (S25 in Fig. 16), the physical layer circuit 2B1 outputs an ident\_done signal and a speed signal (speed signal (.max)) indicative of a maximum data transfer speed of the physical layer circuit 2B1 to the parent port (S26 in Fig. 16 and P26 in Fig. 17). On condition that the setting of the data transfer speed is finished on receipt of the speed signal (speed signal 1) from the personal computer 103 and a self\_ID packet (self\_ID packet n + 1) from the personal computer 103 is received by the parent port (S27 in Fig. 16 and P26 and P27 in Fig. 17), the self-identifying operation is finished.

The case where the device to which the bus analyzer 1B is connected is the digital camera 103 will be described by referring to Figs. 16 and 18. After completion of the tree-identifying operation in Fig. 16, an idle state (S21 in Fig. 16) in which no signal is outputted from each of the ports is derived. After that, on condition that no IDs

(IDs = 0 to n - 1) are assigned to devices in a lower hierarchy layer ("!child\_ID\_complete" in Fig. 16) (S28 in Fig. 16), the physical layer circuit 2B1 in the bus analyzer 1B including the root outputs the self\_ID\_grant signal to the child port (S29 in Fig. 16 and P28 in Fig. 18). The digital camera 103 which has received the self\_ID\_grant signal gives the ID of 0 to itself and outputs a self\_ID packet (self\_ID packet 0). Under condition that the self\_ID\_grant signal is received by the child port (S30 in Fig. 16), the physical layer circuit 2B1 stops outputting (S31 in Fig. 16 and P29 in Fig. 18).

Under condition that an ident\_done signal from the digital camera 103 is received and child\_ID\_complete is set (S32 in Fig. 16), the speed signal (speed signal 1) from the digital camera 103 is received and a speed signal (.max) indicative of the maximum data transfer speed is outputted to the same port (S33 in Fig. 16 and P30 in Fig. 18). The state is finished when a timeout of the speed signal output occurs (S35 in Fig. 16) and the physical layer circuit 2B1 enters an idle state (S21 in Fig. 16).

On set of the child\_ID\_complete (S22 in Fig. 16), the physical layer circuit 2B1 including the root outputs a self\_ID packet (self\_ID packet 1) to the child port, gives the ID of 1 to itself, and increments the ID counter by one (S23 in Fig. 16 and P31 in Fig. 18).

The sequence is repeated until the ID counter becomes n (S24 in Fig. 16) and IDs of 2 to n are sequentially set subsequent to the ID of 1 in the physical layer circuit 2B1 (P32 and P33 in Fig. 18).

When the physical layer circuit 2B1 includes a root, on condition that the ID counter becomes n (S35 in Fig. 16), the self-identifying operation is finished and the initializing sequence is completed (P34 in Fig. 18).

The state transition from S21 to S31 upon receipt of the self\_ID packet by the parent port in Fig. 16 indicates the function of performing

the self-identifying operation without outputting the self\_ID\_grant signal. The state transition from S31 to S21 represents the process of finishing the self\_ID packet without receiving the ident\_done signal.

As described above, since the bus analyzer 1B can simulate the group of devices having a plurality of IDs, even in the case of analyzing or investigating the system in which  $(n + 1)$  devices having IDs of 0 to  $(n)$  (or  $n$  devices having IDs of 1 to  $n$ ) are connected to the personal computer 103 having the ID of  $(n + 1)$  (or the digital camera 103 having the ID of 0), it is unnecessary to actually construct a system to be analyzed by preparing  $(n + 1)$  devices having IDs of 0 to  $n$  (or  $n$  devices having IDs 1 to  $n$ ). By connecting the bus analyzer 1B and the personal computer 103 (or digital camera 103) via the serial bus B3, the physical layer circuit 2B1 in the bus analyzer 1B interfaces with the bus B3 in a state where a plurality of IDs are assigned to the physical layer circuit 2B1, and data is analyzed by the data analyzing circuit 3 at a post stage. Only by connecting a device to be analyzed to the bus analyzer 1B, an environment in which a plurality of devices are connected can be created, and the device connected to the serial bus B3 can be easily and certainly analyzed or investigated.

A second configuration example of the second embodiment of the invention will now be described by referring to Figs. 19 to 25. In the configuration example, three actual devices of a personal computer 104 having an ID of  $(n + 1)$  (or a DVD drive 104 having an ID of 2), a hard disk drive 105 having an ID of 1, and the digital camera 102 having an ID of 0 are included, and the bus analyzer 1B simulates the group of other devices. The configuration example is intended for analyzing or investigating responses from the three devices connected on serial buses B4, B5 and B6. It is, however, unnecessary to prepare  $(n - 1)$  devices having IDs of 2 to  $n$  (or  $(n - 2)$  devices having IDs

of 3 to n) for the analysis or investigation. By connecting the bus analyzer 1B and the three devices, an environment for analysis can be created. The bus analyzer 1B interfaces with the IEEE 1394 buses B4, B5 and B6 in the physical layer circuit 2B2 in the bus analyzer 1B, and realizes data communications with the buses B4, B5 and B6. The received data is analyzed by the data analyzing circuit 3 at a post stage. Since the topology is constructed in the interface by assigning a plurality of IDs to the physical layer circuit 2B2 as will be described hereinlater, the environment in which a plurality of devices except for the three actual devices are connected can be created.

The initializing sequence of the IEEE 1394 bus in the second configuration example of the second embodiment in Fig. 19 will be described hereinbelow. Since the tree structure of the topology is determined in conformity with the IEEE 1394 standard, description of the tree-identifying operation of determining the tree structure of the topology is omitted here. A self-identifying operation of assigning IDs to devices will be described. Fig. 20 is a state transition diagram showing the self-identifying operation in the second configuration example of the second embodiment in Fig. 19. Figs. 21 to 23 show an operation sequence in the case where there is a device having high arbitration (the personal computer 104 having the ID of (n + 1) in Fig. 19) except for the bus analyzer 1B. Figs. 24 and 25 show an operation sequence in the case where the arbitration of the bus analyzer 1B is high.

First, the case where there is the personal computer 104 (ID = n + 1) as a device having high arbitration except for the bus analyzer 1B will be described with reference to Figs. 20 to 23. In Fig. 20, after completion of the tree-identifying operation, an idle state (S41 in Fig. 20) in which no signal is outputted from each of the ports is derived. After that, the personal computer 104 as a root in the

topology outputs the self\_ID\_grant signal to its child port. On condition that the self\_ID\_grant signal is received by the parent port (S42 in Fig. 20 and P41 in Fig. 21), the physical layer circuit 2B2 in the bus analyzer 1B which is not a root outputs a self\_ID\_grant signal to the child port (lowest\_unidentified\_child port) to which the digital camera 102 having no ID is connected, and outputs only a data\_prefix signal to the other child ports (S43 in Fig. 20 and P41 in Fig. 21).

The digital camera 102 which has received the self\_ID\_grant signal sets the ID of 0 to itself and outputs a self\_ID packet (self\_ID packet 0). On condition that the self\_ID packet is received by the lowest\_unidentified\_child port in the physical layer circuit 2B2 (S44 in Fig. 20), the physical layer circuit 2B2 stops outputting the self\_ID\_grant signal and transfers the self\_ID packet (self\_ID packet 0) from the digital camera 102 to each of the ports (S45 in Fig. 20 and P42 in Fig. 21).

On condition that an ident\_done signal from the digital camera 102 is received and child\_ID\_complete is set (S46 in Fig. 20), the physical layer circuit 2B2 receives a speed signal (speed signal 2) from the digital camera 102 and outputs a speed signal (speed signal (.max)) indicative of the maximum data transfer speed of the physical layer circuit 2B2 itself to the same port (S47 in Fig. 20 and P43 in Fig. 21). The state is finished when a timeout of the output of the speed signal occurs (S48 in Fig. 20), and the physical layer circuit 2B2 enters an idle state (S41 in Fig. 20).

When the self\_ID\_grant signal from the personal computer 104 as a root is received by the parent port (S42 in Fig. 20 and P44 in Fig. 21), the self\_ID\_grant signal is outputted to the child port (lowest\_unidentified\_child port) to which the hard disk drive 105 having no ID is connected and a data\_prefix signal is outputted to the other child ports (S43 in Fig. 20 and P44 in Fig. 21).

The hard disk drive 105 which has received the self\_ID\_grant signal sets the ID of 1 to itself and outputs a self\_ID packet (self\_ID packet 1). On condition that the self\_ID packet is received by the lowest\_unidentified\_child port in the physical layer circuit 2B2 (S44 in Fig. 20), the physical layer circuit 2B2 stops outputting the self\_ID\_grant signal and transfers the self\_ID packet (self\_ID packet 1) from the hard disk drive 105 to each of the ports (S45 in Fig. 20 and P45 in Fig. 22).

Further, on condition that the ident\_done signal from the hard disk drive 105 is received and the child\_ID\_complete is set (S46 in Fig. 20), a speed signal (speed signal 3) from the hard disk drive 105 is received and a speed signal (speed signal (.max)) indicative of a maximum data transfer speed of the physical layer circuit 2B2 is outputted to the same port (S47 in Fig. 20 and P46 in Fig. 22). This state is finished when a timeout of the speed signal output occurs (S48 in Fig. 20) and the physical layer circuit 2B2 enters an idle state (S41 in Fig. 20).

At this time, the setting of IDs to devices to which IDs lower than that of the physical layer circuit 2B2 are given is finished and all the child\_ID\_complete is set. Consequently, the routine advances to a sequence of assigning IDs to a group of devices to be simulated by the physical layer circuit 2B2. In a manner similar to the foregoing case, when the self\_ID\_grant signal from the personal computer 104 is received by the parent port (S42 in Fig. 20 and P47 in Fig. 22), since the lowest\_unidentified\_child port does not exist, without outputting the self\_ID\_grant signal, a data\_prefix signal is outputted to all the child ports (S43 in Fig. 20).

Since all the child\_ID\_complete is set at this time point (S49 in Fig. 20), a self\_ID packet (self\_ID packet 2) is outputted to all the ports, the ID counter is incremented by one (S50 in Fig. 20 and

P47 in Fig. 22) and the ID of 2 is assigned to itself.

The sequence is repeated until the ID counter becomes n (S51 in Fig. 20), and IDs from 3 to n are sequentially set subsequent to the ID of 2 to the physical layer circuit 2B2 (P48 in Fig. 22).

On condition that the ID counter becomes n in the physical layer circuit 2B2 which does not include the root (S52 in Fig. 20), the physical layer circuit 2B2 outputs a speed signal (speed signal (.max)) indicative of the maximum data transfer speed of the physical layer circuit 2B2 together with an ident\_done signal to the parent port (S53 in Fig. 20 and P49 in Fig. 23). On condition that the setting of the data transfer speed is finished on receipt of the speed signal (speed signal 1) from the personal computer 104 and the self\_ID packet (self\_ID packet n + 1) from the personal computer 104 is received by the parent port (S54 in Fig. 20 and P49 and P50 in Fig. 23), the self-identifying operation is finished thereby completing the initializing sequence.

The operation of transferring the self\_ID packet (self\_ID packet n + 1) to the child port in P50 in Fig. 23 is performed in normal arbitration.

A case where the physical layer circuit 2B2 in the bus analyzer 1B has high arbitration and includes a root will now be described with reference to Figs. 20, 24 and 25. After completion of the tree-identifying operation in Fig. 20, an idle state in which there is no signal output from each of the ports is obtained (S41 in Fig. 20). After that, on condition that the physical layer circuit 2B2 includes a root (S42 in Fig. 20), the self\_ID\_grant signal is outputted to the lowest\_unidentified\_child port to which the digital camera 102 having no ID is connected and the data\_prefix signal is outputted to the other child ports (S43 in Fig. 20).

The digital camera 102 which has received the self\_ID\_grant signal sets the ID of 0 to itself and outputs a self\_ID packet (self\_ID

packet 0). On condition that the lowest\_unidentified\_child port in the physical layer circuit 2B2 receives the self\_ID packet (S44 in Fig. 20), the physical layer circuit 2B2 stops outputting the self\_ID\_grant signal and transfers the self\_ID packet (self\_ID packet 0) from the digital camera 102 to each of the ports (S45 in Fig. 20 and P51 in Fig. 24).

Further, on condition that the ident\_done signal from the digital camera 102 is received and child\_ID\_complete is set (S46 in Fig. 20), the speed signal (speed signal 1) from the digital camera 102 is received, and the speed signal (speed signal (.max)) indicative of the maximum data transfer speed of the physical layer circuit 2B2 is outputted to the same port (S47 in Fig. 20 and P52 in Fig. 24). This state is finished when a timeout of the speed signal output occurs (S48 in Fig. 20), and the physical layer circuit 2B2 enters an idle state (S41 in Fig. 20).

Similar sequences are sequentially repeated, the ID of the hard disk drive 105 is set to 1, and the ID of the DVD 104 is set to 2 (P53 and P54 in Fig. 24 and P55 and P56 in Fig. 25). Then the physical layer circuit 2B2 returns to the idle state (S41 in Fig. 20).

Since setting of IDs to devices to which IDs lower than that of the physical layer circuit 2B2 are given is finished and a state in which all the child\_ID\_complete is set is obtained, the program progresses to a sequence of assigning IDs to a group of devices simulated by the physical layer circuit 2B2.

On condition that the physical layer circuit 2B2 includes a root (S42 in Fig. 20), the physical layer circuit 2B2 enters the state of S43 in Fig. 20. Since the lowest\_unidentified\_child port does not exist at this time point, in a state where no self\_ID\_grant signal is outputted, the data\_prefix signal is outputted to all the child ports (S43 in Fig. 20). On condition that all the child\_ID\_complete has been set (S49 in Fig. 20), the physical layer circuit 2B2 outputs the self\_ID

packet (self\_ID packet 3) to all the ports, increments the ID counter by one (S50 in Fig. 20 and P57 in Fig. 25), and assigns the ID of 3 to itself.

By repeating the sequence until the ID counter becomes n (S51 in Fig. 20), the IDs of 4 to n are sequentially set subsequent to the ID of 3 in the physical layer circuit 2B2 (P58 in Fig. 25).

On condition that the ID counter becomes n in the physical layer circuit 2B2 including a root (S55 in Fig. 20), the self-identifying operation is finished and the initializing sequence is completed.

The state transition from S41 to S45 in association with reception of the self\_ID packet by the parent port in Fig. 20 represents the function of performing the self-identifying operation without outputting the self\_ID\_grant signal. The state transition from S45 to S41 indicates a process of finishing the self\_ID packet without receiving the ident\_done signal.

As described above, in the second configuration example of the second embodiment of the invention, the three real devices of the personal computer 104 having the ID of  $(n + 1)$  (or the DVD drive 104 having the ID of 2), the hard disk drive 105 having the ID of 1, and the digital camera 102 having the ID of 0 are included. The group of other devices can interface with the serial buses B4, B5 and B6 in a state where a plurality of IDs are assigned by the physical layer circuit 2B2 in the bus analyzer 1B. It is therefore unnecessary to prepare  $(n - 1)$  devices having IDs of 2 to n (or  $(n - 2)$  devices having IDs 3 to n). By connecting the three devices to the bus analyzer 1B, an environment desired to be analyzed can be created. The three devices connected can be therefore easily and certainly analyzed or investigated.

As described above in detail, in the bus analyzer 1A according to the first embodiment, even when the bus analyzer 1A is inserted

between the serial buses B1 and B2 of the system to be analyzed, without assigning an ID in the bus initializing sequence after the insertion by the physical layer circuit 2A, a physical interface can be constructed between the buses B1 and B2. Consequently, without changing the topology environment of the system to be analyzed comprising the personal computer 101 and the digital camera 102 connected to the buses B1 and B2 and exerting an influence on the responses on the buses B1 and B2, the bus analyzer 1A as a serial bus interface device can be connected. Since the physical layer circuit 2A of the inserted bus analyzer 1A constructs a physical interface with the buses B1 and B2, data on the buses B1 and B2 can be detected and the states of the buses B1 and B2 can be analyzed or investigated.

Further, as for the bus analyzers 1A1 to 1A6 of the first to six modifications of the first embodiment, the first modification has the data storing circuit 4 for storing data on the serial buses B1 and B2. Consequently, the bus analyzer 1A1 constructs a physical interface with the serial buses B1 and B2 without being given an ID, and can obtain data on the buses B1 and B2 without changing the topology environment of the system to be analyzed and without exerting an influence on operations. The data can be directly analyzed and, moreover, can be properly stored/read to/from the data storing circuit 4 in accordance with a control signal from the data analyzing circuit 31. A series of continuous data or data in a specific sequence can be therefore collected and analyzed, and can be analyzed by comparison with data obtained from the buses B1 and B2. Thus, analysis can be efficiently carried out.

In the second modification, the bus analyzer 1A2 has, in addition to the data storing circuit 4, the data condition detecting circuit 5 and data received by the physical layer circuit 2A2 is monitored. When data matching the predetermined condition is detected, a trigger

signal is outputted. Consequently, the received data can be stored in the data storing circuit 4 in response to the output of the trigger signal. By setting the data condition of outputting the trigger signal in a specific data sequence, each time the same sequence is generated, data can be stored. Analysis suitable for the case where a malfunction occurs on a specific condition can be performed.

Further, in the third modification, in addition to the data analyzing circuit 33, the data transfer control circuit 6 is provided. Data to be transmitted can be transferred as it is to the physical layer circuit 2A3 or formed in packets compatible with the protocol on the serial buses B1 and B2 and the packets can be transferred to the physical layer circuit 2A3. Thus, responses on the buses B1 and B2 to a predetermined data packet can be certainly recognized.

In the fourth modification, in addition to the data analyzing circuit 34 and the data transfer control circuit 6, the data storing circuit 4 is provided. Consequently, data to be transmitted can be prestored. The data can be transferred as it is to the physical layer circuit 2A4 or formed in packets compatible with the protocol on the serial buses B1 and B2 and the packets can be transferred to the physical layer circuit 2A4. By storing plural data, a response on the buses B1 and B2 to each data packet can be recognized. By arbitrarily setting a sequence of transmitting a data packet, a response to the specific packet sequence or a response in a state where the communications on the buses B1 and B2 are congested can be recognized.

In the fifth modification, in addition to the data analyzing circuit 35, the data storing circuit 4 and the data transfer control circuit 6, the data transmission condition detecting circuit 7 is provided. The data received by the physical layer circuit 2A5 is monitored and, when data matching the predetermined condition is detected, a trigger signal is outputted. The data to be transmitted

which is stored in the data storing circuit 4 can be transferred as it is to the physical layer circuit 2A5 or formed in packets compatible with the protocols on the serial buses B1 and B2 and the packets can be transferred to the physical layer circuit 2A5. By setting the data obtaining condition of outputting the trigger signal in a specific data sequence, each time the same sequence is generated, predetermined data stored in the data storing circuit 4 can be transmitted. Consequently, the same response can be always sent back to the specific data sequence and thus the modification is optimum to recognize the responses on the buses B1 and B2.

In the sixth modification, the bus analyzer 1A6 comprises the pair of ports, the receiving circuit 11, the data converting circuit 10 for converting received data, and the transmitting circuit 12 for forming packets from the transformed data and has a construction in which ports of transmission/reception are switched by the selector 13. On the basis of the analysis of received data by the data bus analyzer 36 or on the basis of data to be transmitted which is stored in the data storing circuit 4, data received by the receiving circuit 11 is converted by the data converting circuit 10, thereby enabling a communication error such as a bit error or a burst error in communications on the serial bus to be simulated. The communication response such as an error correcting function can be therefore recognized.

The bus analyzer 1B according to the second embodiment interfaces with the serial bus B3 in a state where a plurality of IDs are assigned to the physical layer circuit 2B1 and can simulate a group of a plurality of devices. In the case of analyzing or investigating a system in which  $(n + 1)$  devices having the IDs of 0 to  $n$  (or  $n$  devices having the IDs 1 to  $n$ ) are connected to the personal computer 103 having the ID of  $(n + 1)$  (or the digital camera 103 having the ID of 0), it is unnecessary

to prepare  $(n + 1)$  devices having the IDs 0 to  $n$  (or  $n$  devices having the IDs 1 to  $n$ ) and actually construct the system to be analyzed. By connecting the bus analyzer 1B and the personal computer 103 (or the digital camera 103) via the bus B3, the environment of the system to be analyzed can be created, and data is analyzed by the data analyzing circuit 3 at a post stage. Thus, the device to be connected to the bus B3 can be easily and certainly analyzed or investigated.

Further, the bus analyzer 1B can have the construction including the three actual devices of the personal computer 104 having the ID of  $(n + 1)$  (or the DVD drive 104 having the ID of 2), the hard disk drive 105 having the ID of 1, and the digital camera 102 having the ID of 0. A plurality of IDs can be assigned to the physical layer circuit 2B2 with respect to the group of other devices. The bus analyzer 1B can construct an interface on the serial buses B4, B5 and B6. In this case as well, it is unnecessary to prepare  $(n - 1)$  devices having the IDs of 2 to  $n$  (or  $(n - 2)$  devices having the IDs 3 to  $n$ ). By connecting the three actual devices, the environment desired to be analyzed can be created. The connected three devices can be easily and certainly analyzed or investigated.

Obviously, the present invention is not limited to the foregoing embodiments but various improvements and modifications are possible without departing from the gist of the invention.

For example, in the embodiments, the bus analyzer 1B according to the second embodiment can obviously have therein functions similar to those of each of the first to six modifications of the bus analyzer 1A in the first embodiment. In this case, with respect to the mode in which the physical layer circuit has therein the data converting circuit for converting received data, since the bus analyzer 1B according to the second embodiment can have multiple ports, a communication error during communication can be transmitted from an

arbitrary port. A response of a device connected to the arbitrary port to an influence of the communication error can be therefore analyzed or investigated.

Although the IEEE 1394 serial bus has been described as an example of the serial bus in the foregoing embodiments, the invention is not limited to the IEEE 1394 serial bus. The invention can be similarly applied to other bus systems capable of automatically assigning an ID to a device to be connected on the bus when the device is inserted or withdrawn.

According to the invention, since an ID is not given to the serial bus interface device to be connected to the serial bus such as the IEEE 1394 serial bus when the serial bus interface device is inserted between devices connected via the serial bus, no influence is exerted on the topology. Since a plurality of IDs can be also assigned, it is unnecessary to prepare all of actual devices to construct a bus system. A bus system can be easily and certainly constructed by simulating some of the devices. The serial bus interface device having the physical layer circuit which is preferably used as a bus analyzer for analyzing and investigating the communicating state of a bus can be provided.